An Algebraic Attack against Augot-Finiasz Cryptosystem

Pierre Loidreau

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An Algebraic Attack against Augot-Finiasz Cryptosystem

Pierre Loidreau

N° 5662

Août 2005

Thème SYM

ISSN 0249-6399 ISRN INRIA/RR--5662--FR+ENG

An Algebraic Attack against Augot-Finiasz Cryptosystem

Pierre Loidreau*

Thème SYM — Systèmes symboliques
Projets Codes

Rapport de recherche n° 5662 — Août 2005 — 13 pages

Abstract: We design an efficient probabilistic attack against the Augot-Finiasz cryptosystem presented at Eurocrypt 2003 enabling an attacker to recover the plaintext of the transmission if one uses the way to reduce the size of the public-key as suggested by the authors. To achieve this we use simple algebraic tools such as the Trace operator. The attacks succeeds in less than 5 minutes on the original parameters.

Key-words: Public-Key cryptosystems, Polynomial Reconstruction, Reed-Solomon codes, Trace operator

* Unité de Mathématiques appliquées, ENSTA

Une attaque algébrique contre le cryptosystème Augot-Finiasz

Résumé : Dans ce rapport, nous décrivons une attaque probabiliste efficace contre le cryptosystème Augot-Finiasz, qui fut présenté à la conférence Eurocrypt 2003. Elle permet à un attaquant de retrouver le texte clair transmis si l'on utilise une des façons de réduire la taille de la clé publique suggérées par les auteurs. A ces fins, nous utilisons des outils algébriques simples, tels que l'opérateur Trace. Sur les paramètres originaux, l'attaque termine en 5 minutes.

Mots-clés : Cryptosystèmes à clé publique, reconstruction de polynômes, codes de Reed-Solomon, opérateur Trace

1 Introduction

One of major problems in public-key cryptography is that the most used cryptosystems all rely on a few specific problems considered to be hard such as the difficulty of factoring integers and computing discrete logarithms. Thus, finding new hard problems and managing to design cryptosystems based on these problems is a promising research field.

The problem we are interested in is the polynomial reconstruction problem (hereafter denoted PR problem) [1], that can be expressed as follows: Given $n,\ k,\ t$ and $(g_i, y_i)_{i=1\ldots n}$, find all the polynomials p of degree less than k such that $p(g_i) = y_i$ for at least t values of i. As stated in [8], this problem is considered to be hard, and therefore provides a solid ground to conceive cryptosystems. This problem is shown to be equivalent to the list-decoding of Reed-Solomon codes.

At EUROCRYPT 2003 Augot and Finiasz were the first to design a public-key cryptosystem (PKC) based upon the PR problem. Basically the original scheme uses as public-key a code-vector of a Reed-Solomon code scrambled by the addition of an error-vector of sufficiently large Hamming weight such that the PR problem is intractable. The conditions were given by Guruswami and Sudan in [7]. Additionally to being an alternative to the existing PKC's based on the factorization or on discrete logarithm problems, the system provided a solution to the reminiscent large key-size of PKC's such as McEliece system or HFE. Namely, the authors proposed in particular a system with a key of 3072 bits for a security of 2^{80} binary operations against the classical attacks such as Information Set decoding and Error-Set decoding [9, 3].

Unfortunately for this nice idea, the system was broken in polynomial-time by Coron. He exploited a linear transformation used in the encryption step, consisting of multiplying the public-key by a random element taken in some large finite field [4, 5]. He adapted the Welch-Berlekamp algorithm [2] and showed how to construct in polynomial time a univariate polynomial of low degree annihilating the random chosen element. Since the number of its roots is upper-bounded by the low degree of the polynomial, the system can be broken in polynomial-time.

In this paper we exhibit a very different attack from Coron's by exploiting algebraic properties of finite fields and properties of the Trace operator. Though of exponential complexity, this attack prove to be very effective on the parameters proposed by the authors. Thus, even if one finds a way to resist Coron's attack, reducing the size of the public-key should not be done in this way.

Although one could state , since Coron already broke the original system in polynomial-time, a new system was presented recently at WCC 2005 using a similar approach. The difference is that this new system establishes its strength on properties of rank metric rather than on Hamming metric [6]. Hence our approach shows that some choices which would allow us to simplify the system in terms of complexity must be avoided.

In the first part of the paper, we introduce the family of Reed-Solomon codes and some of its properties constituting the heart of the system. In a second part we present the original encryption scheme, and the variant enabling to reduce the size of the public-key. Then we present the trace-operator and the result of its action on RS codes. In the final part we

design the attack using the trace operator, and we prove that the attack effectively works in probabilistic polynomial-time. With the parameters proposed, this attack implemented thanks to the MAGMA software succeeds in less than 5 minutes on a Processor 1Ghz.

2 Reed-Solomon Codes

In this section we present algebraic tools used in the design of the system and further used to analyse its security. Reed-Solomon codes (Hereafter denoted RS codes) are well-known optimal codes and are closely related to the rings of univariate polynomials over finite fields.

Let q be an integer which is a power of a prime. We consider the finite field $\mathrm{GF}(q)$ with q elements, unique up to isomorphism, and the finite field $\mathrm{GF}(q^u)$, where u is some non-zero integer. With these notations, the finite field $\mathrm{GF}(q)$ is a subfield of $\mathrm{GF}(q^u)$.

Let us consider the labelled set

$$S = (g_1, g_2, \ldots, g_n)$$

of n distinct elements chosen in $\mathrm{GF}(q^u)$. Let ev_S denote the evaluation of polynomials on the elements of S, *i.e.*

$$\mathrm{ev}_S \left\{ \begin{array}{ccc} \mathrm{GF}(q^u)[X] & \longrightarrow & \mathrm{GF}(q^u)^n \\ p(X) & \longmapsto & (p(g_1), p(g_2), \ldots, p(g_n)), \end{array} \right.$$

where $\mathrm{GF}(q^u)[X]$ is the set of univariate polynomials with coefficients in $\mathrm{GF}(q^u)$.

Definition 1 *The Reed-Solomon code of dimension k and of support S is*

$$RS_k(S) = \{\mathrm{ev}_S(f) \mid f \in GF(q^u)[X] \text{ and } \deg(f) < k\}.$$

With this definition $RS_k(S)$ can be viewed as the set of the vectors formed by the evaluation of all the polynomials of degree less than k on the elements of S.

The code $RS_k(S)$ can correct up to $\lfloor (n-k)/2 \rfloor$ corrupted positions in polynomial time by using a Berlekamp-Massey or Welch-Berlekamp algorithm [2].

Definition 2 (Parity-check matrix of RS code)
A parity-check matrix for the code $RS_k(S)$ of length n is a $(n-k) \times n$ matrix H of full rank such that for any vector $m \in RS_k(S)$ we have $Hm^t = 0$.

With such a description, RS codes are evaluation codes of polynomials of fixed degree on elements taken in a finite field. The difficult problem on which the security of the system relies is the Polynomial reconstruction problem that can be stated as follows.
Polynomial Reconstruction problem:

Given integers n, k, t and g_i, y_i elements of some field, find a polynomial $p(X)$ of degree less than k such that $p(g_i) = y_i$ on at least t values of i.

When we consider distinct elements over some finite field, this problem is strictly equivalent to the decoding problem of RS codes.

3 Original system

The system conceived by Augot and Finiasz uses as public-key a vector of $RS_k(S)$ scrambled by an error-vector of sufficiently large weight such that the polynomial reconstruction problem is intractable.

The original system is the following:

- *Known Parameters:* A finite field $\mathrm{GF}(q^u)$, integers $n,\ k,\ W,\ w$, a set $S = (g_1, \ldots, g_n)$ of n distinct elements of $\mathrm{GF}(q^u)$.
- *Key generation:* Alice chooses randomly a monic polynomial $p(X)$ of degree $k-1$ over $\mathrm{GF}(q^u)$, and computes $c = \mathrm{ev}_S(p)$. The vector c belongs thus to $RS_k(S)$. Then she generates randomly a vector $E = (E_1, \ldots, E_n) \in \mathrm{GF}(q^u)^n$ with exactly W non-zero coordinates. The public-key consists of $K = c + E$. Both c and E remain secret.
- *Encryption:* Bob wants to send a message $m_0 = (m_{0,0}, \ldots, m_{0,k-2})$ of length $k-1$ over $\mathrm{GF}(q^u)$ to Alice. First he transforms m_0 into the polynomial $m_0(X) = \sum_{i=0}^{k-2} m_{0,i} X^i$ and computes $m = \mathrm{ev}_S(m_0)$. Then he picks up randomly $\alpha \in \mathrm{GF}(q^u)$, and a vector e of length n with w non-zero coordinates. The ciphertext Bob sends to Alice is
$$y = m + \alpha K + e.$$
- *Decryption:* Alice first shortens the ciphertext on the non-zero positions of E by removing all coordinates corresponding to the non-zero positions of E. She obtains $\overline{y} = \overline{m} + \alpha\overline{c} + \overline{e}$, where the overlining denotes shortening the vectors. Hence we obtain $\overline{K} = \overline{c}$. A shortened RS code is still a RS code. Therefore by one decoding step in the shortened RS code, Alice recovers $\overline{m} + \alpha\overline{c}$. Since $\overline{m} = \mathrm{ev}_{\overline{S}}(m_0)$ and $\alpha\overline{c} = \mathrm{ev}_{\overline{S}}(\alpha p)$ and since the degree of m_0 is less than the degree of p, Alice first recovers $Q(X) = m_0(X) + \alpha p(X)$ by interpolation. Then, by considering that the highest coefficient of Q is α, she recovers α, and finally m_0.

In the design of the system the size of the public-key is equal to $n \times u\log_2(q)$ bits. Now the size of the public-key has to be quite large since the security of the cryptosystem implies that

- the PR problem must be intractable, so that any attempt to break the system by recovering the secret parameters is infeasible;
- the system must resist known attacks such as information-set decoding and error-set decoding attacks. These attacks are standard procedures which recover the error-vector e, by trying to decode any intercepted ciphertext in a designed code. For further references to these procedures see [3, 9].

Therefore, the authors proposed to take: $q = 2^{12},\ u = 7$, that is $\mathrm{GF}(q^u) = \mathrm{GF}(2^{84})$, $n = 1024$, $W = 74$. With such parameters, the size of the public key $c + E$ is equal to

$84 \times 1024 = 84$ kbits. It is large but remains smaller than the public-key needed in McEliece cryptosystem.

Then by considering that this key-size is a bit too large for practical use, they proposed to reduce the key-size by considering the set S, support of the code, with elements in the subfield $\mathrm{GF}(q)$ of $\mathrm{GF}(q^u)$.

The only different thing is that the support S is formed of elements of $\mathrm{GF}(q)$, and that the polynomial p is chosen with coefficients in $\mathrm{GF}(q)$ rather than in $\mathrm{GF}(q^u)$. Thus $c = ev_S(p)$ has also coefficients in $\mathrm{GF}(q)$ and the size of the public-key can be made as low as $n \times \log_2(q)$. The authors proposed to choose $\mathrm{GF}(q^u) = \mathrm{GF}(2^{84})$ and $\mathrm{GF}(q) = \mathrm{GF}(2^{12})$. The size of the public-key is decreased down to 12 kbits.

Furthermore, by taking c in the subfield subcode over $\mathrm{GF}(8)$ of $RS_k(S)$, the size of the public-key can be made as low as to $3 \times 1024 = 3072$ bits, together with keeping a sufficient security against decoding attacks.

In the next sections we show that such a reduction induces an algebraic attack against the system. For the proposed parameters, this attack is extremely fast.

4 The Trace operator

This section is dedicated to the description of the Trace operator and some of its properties used in the design of our attack.

The Trace operator is a classical tool of algebraic theory over the finite fields. The finite field $\mathrm{GF}(q^u)$ can be viewed as a u-dimensional vector space over $\mathrm{GF}(q)$, that is there exists a set $\gamma_1, \ldots, \gamma_u$ of u elements of $\mathrm{GF}(q^u)$ such that any element α of $\mathrm{GF}(q)$ can be uniquely written under the form

$$\alpha = a_1\gamma_1 + \cdots + a_u\gamma_u,$$

where the a_i's lie in the small field $\mathrm{GF}(q)$. The function of $\mathrm{GF}(q^u)$ into $\mathrm{GF}(q^u)$ defined by

$$x \mapsto x^q$$

is $\mathrm{GF}(q)$-linear, since for all $a \in \mathrm{GF}(q)$, and for all $x, y \in \mathrm{GF}(q^u)$, we have $(a(x+y))^q = ax^q + ay^q$. We define now the trace operator:

Definition 3
The trace operator of GF(q^u) into GF(q) is defined by

$$\forall x \in GF(q^u),\ \ Tr(x) = x + x^q + \cdots + x^{q^{u-1}},$$

It is easy to see that since $\mathrm{Tr}(x)^q = \mathrm{Tr}(x)$, the trace operator takes its values in the subfield $\mathrm{GF}(q)$ of $\mathrm{GF}(q^u)$. Being the sum of linear functions, it is also a linear function. Furthermore, for all element a taken in the subfield $\mathrm{GF}(q)$ it satisfies the relation:

$$\mathrm{Tr}(ax) = a\mathrm{Tr}(x),\ \forall x \in \mathrm{GF}(q^u).$$

This property is not true in general for a taken in $\mathrm{GF}(q^u)$.

Proposition 1

Let $\gamma_1, \ldots, \gamma_u$ be any basis of GF(q^u) over GF(q), then there exist a unique basis $\beta_1, \ldots, \beta_u$ of GF(q^u) over GF(q) such that

$$Tr(\gamma_i \beta_j) = \delta_{i,j},$$

where $\delta_{i,j}$ denotes the Kronecker symbol.

Such a basis can be easily computed see [10].

The number of non-zero elements α in $\mathrm{GF}(q^u)$ such that $\mathrm{Tr}(\alpha) = 0$ is huge. More precisely,

Proposition 2

The set of elements $\alpha \in GF(q^u)$, such that $Tr(\alpha) = 0$ is a $u-1$ dimensional vector-space over GF(q).

Hence the number of such elements is equal to q^{u-1}.

The action of the Trace operator on a vector $c = (c_1, \ldots, c_n)$ of length n is defined by $\mathrm{Tr}(c) = (\mathrm{Tr}(c_1), \ldots, \mathrm{Tr}(c_n))$.

The fundamental proposition linking the Trace operator and RS codes is the following:

Proposition 3

Let $S = (g_1, \ldots, g_n)$ where $g_i \in GF(q)$ for all $i = 1, \ldots, n$. Then for all $c = ev_S(p)$ with $p(X) = \sum_{i=0}^{k-1} p_i X^i$, we have

$$Tr(c) = ev_S(P),$$

where $P(X) = \sum_{i=0}^{k-1} Tr(p_i) X^i$. As a consequence $RS_k(S)$ is stable under the action of the Trace operator.

PROOF

Since $c = ev_S(p)$, for every component c_j of c, we have $c_j = p(g_j) = \sum_{i=0}^{k-1} p_j g_i^j$. Since $g_j \in \mathrm{GF}(q)$ and by $\mathrm{GF}(q)$-linearity of the trace operator, for all j we have $\mathrm{Tr}(c_j) = \sum_{i=0}^{k-1} \mathrm{Tr}(p_i) g_j^i = \mathrm{ev}_S(P)_j$, where $\mathrm{ev}_S(P)_j$ denotes the jth component of the vector $\mathrm{ev}_S(P)$.

◇

5 Trace-based attacks

In the original system, Augot and Finiasz proposed a solution to reduce the public-key size from some 80 kbits down to 3096 bits, together with keeping a sufficient strength against classical decoding attacks. This reduction can be described in two steps.

1. First they construct the RS code from the system in a particular way. Namely, it satisfies the hypotheses of Proposition 3, the support being taken in some subfield $\mathrm{GF}(q)$ of the original field $\mathrm{GF}(q^u)$. With such an approach they can take words of the RS code with coefficients in $\mathrm{GF}(q)$ rather than in $\mathrm{GF}(q^u)$. Then they add an error-vector E equally with coefficients in $\mathrm{GF}(q)$. The size of the public-key is thus reduced u times without decreasing the security against decoding attacks.

2. Second, they take the private key in some subfield subcode of the RS code. This decreases the dimension of the code, but it is possible to keep a sufficient security with a key size of 3096 bits.

This approach however is dangerous and gives grip to algebraic attacks. The attack we propose deals with the first point. Since the second point depend on the first one we show that it is not possible to take the public-key with coefficient in a small field without introducing a very important weakness.

The attack we construct relies on algebraic properties of the trace operator exposed in the previous section. Given an intercepted ciphertext, this attack enables an eavesdropper to recover the non-zero positions of the error-vector e which has been added for a scrambling effect.

It is a probabilistic-time algorithm whose complexity depends on the field in which the coefficients of the public-key are taken. If the field is small then the attack is very fast. Basically one has to enumerate the elements of the field GF(q) and, complete some polynomial-time tests on each of them. Whenever the public-key is taken with coefficients in GF(2^{10}), it takes less than a minute to recover the plaintext from an intercepted ciphertext.

First we show that it is enough to recover the positions of the error-vector e to recover the plaintext in polynomial-time. Then we describe the algorithm doing this, and study its complexity. We finally prove that it always returns the right error-positions.

5.1 Ground of the attack

Augot-Finiasz system suffers from the fact that, as soon as one recovers any of the secret parameters, either $\alpha \in \mathrm{GF}(q^n)$ or the error-vector e, then the plaintext can be found in polynomial-time. As a matter of fact Coron proposed a polynomial-time attack recovering first α, then e and finally the whole plaintext.

However, knowing e is too much of information. Indeed it is enough to get the w positions where e is non-zero. Suppose the eavesdropper Eve intercepts a ciphertext $y = m + \alpha K + e$ and manages to recover the non-zero positions of e. She can compute the vector $\widetilde{y}$ of length $n - w$ obtained from the ciphertext y by suppressing the w non-zero positions of e. This gives

$$\widetilde{y} = \widetilde{m} + \alpha \widetilde{K},$$

where the unknowns are

- $\widetilde{m}$ – the plaintext m from which one has suppressed the w positions corresponding to the w non-zero positions of e.
- The element $\alpha \in \mathrm{GF}(q^u)$.

Let $\widetilde{S}$ bet the set of elements obtained from the support S by suppressing the w elements corresponding to the w non-zero positions of e. In this way we get $\widetilde{m} \in RS_k(\widetilde{S})$.

Let H be a parity-check matrix of $RS_k(\widetilde{S})$ – see definition 2. By computing $\widetilde{y}H$, one obtains

$$\widetilde{y}H = \underbrace{\widetilde{m}H}_{0} + \alpha\widetilde{K}H = \alpha\widetilde{K}H$$

Since everything is known in this equation except the element α, Eve can recover α by doing the division between the ith coordinate of the vector $\widetilde{y}H$ and the ith coordinate of the vector $\widetilde{K}H$ that is non-zero.

After that she gets $\widetilde{m}$ by computing

$$\widetilde{m} = \widetilde{y} - \alpha\widetilde{K}.$$

From $\widetilde{m} = ev_{\widetilde{S}}(m_0)$, Eve gets m_0 by interpolating the polynomial m_0 of degree $k-1$ on the $n - w > k$ elements of $\widetilde{S}$.

All these operations are polynomial and low in complexity. Therefore we have shown that getting the non-zero positions of e is enough to break the system.

5.2 Recovering the positions of e

In this section we show that, under the modifications proposed by the authors of the original scheme, the non-zero positions of e can be obtained with a very simple probabilistic polynomial-time algorithm.

Let us consider the finite field $\mathrm{GF}(q^u)$. Let the support S take all its elements in the field $\mathrm{GF}(q) \subset \mathrm{GF}(q^u)$ and let $c = \mathrm{ev}_S(p) \in RS_k(S)$, where $p(X)$ has coefficients in $\mathrm{GF}(q)$. The error-vector E part of the secret is taken over $\mathrm{GF}(q)$ with W non-zero coordinates. Hence the public-key $K = c + E$ of the system has coefficients in $\mathrm{GF}(q)$. Let Tr denote the trace operator of $\mathrm{GF}(q^u)$ into $\mathrm{GF}(q)$, *i.e.* for all element γ in $\mathrm{GF}(q^u)$, one has

$$\mathrm{Tr}(\gamma) = \gamma + \gamma^q + \cdots + \gamma^{q^{u-1}}.$$

Suppose Eve intercepts the ciphertext $y = m + \alpha K + e$. For any element γ in $\mathrm{GF}(q^u)$, she can compute $\mathrm{Tr}(\gamma y)$ in linear time. This leads to the following equation

$$\mathrm{Tr}(\gamma y) = \mathrm{Tr}\left(\gamma(m + \alpha K + e)\right).$$

By linearity of the trace operator one has

$$\mathrm{Tr}(\gamma y) = \mathrm{Tr}(\gamma m) + \mathrm{Tr}(\gamma\alpha K) + \mathrm{Tr}(\gamma e).$$

However, by construction of the system with reduced key-size, the public-key K has coefficient in the field $\mathrm{GF}(q)$. By properties of the trace operator, any element of $\mathrm{GF}(q)$ can be extracted from the trace operator. Thus $\mathrm{Tr}(\gamma\alpha K) = \mathrm{Tr}(\gamma\alpha)K$, and

$$\mathrm{Tr}(\gamma y) = \mathrm{Tr}(\gamma m) + \mathrm{Tr}(\gamma\alpha)K + \mathrm{Tr}(\gamma e).$$

Now, by construction of the RS code, $\mathrm{Tr}(\gamma m)$ lies in $RS_{k-1}(S)$. Namely, m is the vector consisting of the evaluation of a polynomial m_0 of degree less than $k-1$ on the elements of S. Hence γm is the evaluation of the polynomial γm_0 over S. The polynomial γm_0 has degree less than $k-1$, thus γm belongs to the code $RS_{k-1}(S)$. By Proposition 3, $RS_{k-1}(S)$ is stable under the action of the Trace operator. Note that since $\mathrm{Tr}(0) = 0$, for any non-zero element $\gamma \in \mathrm{GF}(q^u)$ the non-zero positions of the vector $\mathrm{Tr}(\gamma e)$ correspond exactly to non-zero positions of e. Thus recovering $\mathrm{Tr}(\gamma e)$ gives us information on the non-zero positions of e.

Suppose Eve knows γ, a non-zero element of $\mathrm{GF}(q^u)$ such that $\mathrm{Tr}(\gamma\alpha) = 0$, then

$$\mathrm{Tr}(\gamma y) = \mathrm{Tr}(\gamma m) + \mathrm{Tr}(\gamma e),$$

where $\mathrm{Tr}(\gamma e)$ has at most w non-zero coordinates, and where $\mathrm{Tr}(\gamma m) \in RS_{k-1}(S)$. Since the number of non-zero positions of $\mathrm{Tr}(\gamma e)$ is less than w, Eve can recover $\mathrm{Tr}(\gamma e)$ by one decoding step.

On the ground of these remarks we have designed the the following algorithm:

Input: Known parameters of the system, an intercepted ciphertext y.
Output: $\gamma \in \mathrm{GF}(q^u)$, $m' \in RS_{k-1}(S)$, e' with at most w non-zero coordinates, such that $\mathrm{Tr}(\gamma y) = m' + e'$.

1. Pick up randomly an element $\gamma \in \mathrm{GF}(q^u)$;
2. Compute $y' = \mathrm{Tr}(\gamma y)$;
3. Try to Decode y' in $RS_{k-1}(S)$;
4. While the decoding of y' in $RS_{k-1}(S)$ fails do
   (a) Pick up randomly an element $\gamma \in \mathrm{GF}(q^u)$;
   (b) Compute $y' = \mathrm{Tr}(\gamma y)$;
5. return γ and the vectors m', e' obtained from the decoding, *i.e.* such that $m' + e' = \mathrm{Tr}(\gamma y)$

Table 1: Algorithm for recovering non-zero positions of the error-vector e

Whenever the algorithm stops, one gets two vectors satisfying $\mathrm{Tr}(\gamma y) = m' + e'$, and e' has at most w non-zero coordinates. Now we have to show that the vector e' returned by the algorithm corresponds exactly to the vector $\mathrm{Tr}(\gamma e)$.

Let m' and e' such that

$$\mathrm{Tr}(\gamma y) = m' + e',$$

where m' belongs to the RS code and e' has at most w non-zero coordinates. Then since $y = m + \alpha K + e$, we have

$$\mathrm{Tr}(\gamma m) + \mathrm{Tr}(\gamma\alpha)K + \mathrm{Tr}(\gamma e) = m' + e'$$

That is, by decomposing $K = c + E$, we finally get

$$\mathrm{Tr}(\gamma m) + \mathrm{Tr}(\gamma\alpha)c - m' = e' - \mathrm{Tr}(\gamma e) - \mathrm{Tr}(\gamma\alpha)E.$$

The right part of the equation is a vector with at most $W + 2w$ non-zero coordinates. Since c belongs to $RS_k(S)$ and since $\mathrm{Tr}(\gamma m)$ and m' belong to $RS_{k-1}(S)$, the left part of the equation is a vector of $RS_k(S)$. Therefore by properties of the RS codes, it is either equal to 0, or has at least $n - k + 1$ non-zero coordinates.

It is not difficult to see that if one wishes to construct a system resisting the known decoding attacks, then the parameters must satisfy $W + 2w \leq n - k$. Therefore in our case this implies that the left part of the equation is equal to zero. By construction, $\mathrm{Tr}(\gamma m)$ and m' correspond to polynomials of degree at most $k-2$, whereas c is the evaluation vector of the polynomial p of degree exactly $k-1$. This implies that $\mathrm{Tr}(\gamma\alpha)p_{k-1} = 0$. Since $p_{k-1} \neq 0$, we have $\mathrm{Tr}(\gamma\alpha) = 0$. We have proved the following proposition.

Proposition 4 (Termination of the algorithm)
Let $y = m + \alpha K + e$ be an intercepted ciphertext from the cryptosystem with reduced public-key. The vector e' and the element $\gamma \in GF(q^u)$ returned by the algorithm satisfy:

- $Tr(\gamma\alpha) = 0$,
- $Tr(\gamma e) = e'$.

5.3 Complexity of the attack

The algorithm for finding non-zero positions of e terminates when one obtains an element $\gamma \in \mathrm{GF}(q^u)$ such that $\mathrm{Tr}(\gamma\alpha) = 0$, where α is not known.

Since the trace operator is a non-zero linear application of $\mathrm{GF}(q^u)$ into $\mathrm{GF}(q)$, its kernel has exactly q^{u-1} elements, meaning that there are exactly q^{u-1} elements of $\mathrm{GF}(q^u)$ with a trace equal to 0. Supposing that γ is picked up with a uniform distribution over $\mathrm{GF}(q^u)$, then $\gamma \mapsto \gamma\alpha$ describes also a uniform distribution over $\mathrm{GF}(q^u)$. Thus, the probability of picking up a non-zero element $\gamma \in \mathrm{GF}(q^u)$ such that $\mathrm{Tr}(\gamma\alpha) = 0$ is exactly equal to

$$P = (q^{u-1} - 1)/q^u \approx 1/q.$$

However there can be non-zero coordinates in e which become 0 under the action of $\mathrm{Tr}(\gamma e)$. Since there are w non-zero independently distributed coordinates, the probability to be in such a case is equal to w/q.

Whenever this happens, one has to run another time the algorithm. If it happens once more one runs it again. After s tries, the probability that we have not found all the non-zero coordinates of e is w/q^s, decreasing exponentially with the number of tries, since all the random variables are independent.

Each try of the algorithm costs at most one decoding in the RS code that is is $\mathrm{O}(n^2)$ operations in $\mathrm{GF}(q)$.

The average complexity of the whole attack enabling to recover the plaintext m_0 from the ciphertext is thus:

- *Finding the non-zero positions of* e : On average it is at most equal to $O(sqn^2)$ bit operations in GF(q).
- *Computing a parity-check matrix of* $RS_k(\widetilde{S})$: $k \times (n-w)$ operations in GF(q^u).
- *Computing* $\widetilde{y}H$ *and* $\widetilde{K}H$: $(n-w) \times (n-k-w)$ operations in GF(q).
- *Computing* α : One multiplication in GF(q^u).
- *Recovering* m_0 *from* $\widetilde{m}$: It is just a Lagrange interpolation over k positions that is $O(k^2)$ operations over GF(q).

This attack was implemented with the parameters given by the authors, that is to know GF(q^u) = GF(2^{84}) and GF(q) = GF(2^{12}). We used the MAGMA language with a CPU speed of 1Ghz. The non-zero positions of e where found in less than 5 minutes. In the appendix, we join the MAGMA code enabling to design the attack.

References

[1] D. Augot and M. Finiasz. A public key encryption scheme bases on the polynomial reconstruction problem. In *EUROCRYPT 2003*, pages 222–233, 2003.

[2] E. R. Berlekamp and L. Welch. Error correction of algebraic block codes. US Patent, Number 4,633,470, 1986.

[3] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, January 1998.

[4] J.-S. Coron. Cryptanalysis of a public-key encryption scheme based on the polynomial reconstruction problem. http://eprint.iacr.org/2003/036/.

[5] J.-S. Coron. Cryptanalysis of a public-key encryption scheme based on the polynomial reconstruction problem. In F. Bao, R. Deng, and J. Zhou, editors, *7th International Workshop on Theory and Practice in Public Key Cryptography, PKC 2004*, volume 2947, pages 14–28. Springer, 2004.

[6] C. Faure and P. Loidreau. A new public-key cryptosystem based on the problem of recontructing p-polynomials. In P. Charpin, O. Ytrehus, and D. Augot, editors, *Proceedings of WCC 2005, March 14–19, Bergen*, March 2005.

[7] V. Guruswami and M. Sudan. Improved decoding of Reed–Solomon and algebraic geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[8] A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of Reed–Solomon codes. In *Proceedings of ICALP 2002*, volume 2380 of *LNCS*, pages 232–243, 2003.

[9] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C. G. Günter, editor, *Advances in Cryptology - EUROCRYPT'88*, volume 330 of *LNCS*, pages 275–280. Springer-Verlag, 1988.

[10] A. J. Menezes. *Applications of Finite Fields.* Kluwer Academic Publishers, 1993.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399